REMARKS/ARGUMENTS

This Amendment is in response to the Office Action dated June 29, 2005. Claims 1-26 are pending. Claims 1-26 are rejected. Claims 1, 12, 16, and 24 have been amended. Accordingly, claims 1-26 remain pending in the present application. A request for extension of time to extend the period for reply for one month from September 29, 2005 to October 29, 2005, is requested herewith.

Claims 1, 12, and 16 have been amended to recite that "at least one of the private and public keys" associated with a software product "is digitally signed by a publisher private key." Claims 1 and 12 have also been amended to recite that "the product private and public keys" are included with an authorization program. Claim 16 is further been amended to recite that the license request is signed using the private key "associated with the product certificate." Claim 24 has been amended to recite that a publisher certificate is "digitally signed by" a certificate authority. Dependent claims, 11, 15, and 23 have been amended to recite "license request," rather than "license."

In the Office Action, the Examiner rejected claims 1-30 and 34-35 under 35 U.S.C. §102(e) as being unpatentable over U.S. Patent No. 6,898,706 (Venkatesan). Applicants respectfully disagree. Anticipation requires that a prior art reference disclose each and every claim element of the claimed invention. It is respectfully submitted that Venkatesan fails to disclose each in every claim element of the independent claims.

The present invention provides a method and system for the delivery of secure software license information to authorize the use of a software product, such as a software program or software resource. The system includes a computer system for executing the software product and an authorizing program that authorizes use of the

software product, and a license server connected to the computer system over a network. A publisher certificate and a product certificate are associated with the software product to be authorized, wherein both the publisher certificate and the product certificate include respective private/public key pairs, and wherein at least one of the product certificate private and public keys is digitally signed by the publisher private key associated with the publisher certificate. In a further embodiment, the publisher certificate is digitally signed by a certificate authority.

When the software product is invoked, the authorizing program generates a license request containing user and product information. The license request is digitally signed by the product private key associated with the product certificate and transmitted to the license server. The license server then generates a license with license terms using data extracted from the license request, digitally signs the license with the publisher private key associated with the publisher certificate, and transmits and the signed license back to a key authority. The key authority generates a license using data extracted from the license request and license terms, validates the license by signing the license with the publisher private key associated with the publisher certificate, and transmits the signed license to the authorizing program. The authorizing program then validates the signed license using the publisher public key, and uses the license terms to control the use of the software product.

Referring now to independent claims 1 and 12, Applicants refer broadly to private and public key pairs, rather than to certificates, which usually include other information besides keys. According to the preferred embodiment, the product key is connected to the publisher key so that only that publisher can allow the product to be authorized. To accomplish this, at least one of the keys of the product is digitally signed by the private

key of publisher. Using the public key of the publisher, the authorization program can verify that the publisher who signed its product public key is the same publisher who signed the license in response to license request.

In independent claims 16, Applicants refer to signed certificates, which have associated private keys. The public keys are part of the certificate. So in this case, the product certificate is digitally signed by the publisher private key. This allows the authorization program to verify that the publisher who signed its product certificate is the same publisher who signed the license.

In independent claim 24, a certificate authority that also has a certificate (i.e., a private/public key pair) is recited. The publisher certificate, which contains the publisher public key, is signed by the certificate authority private key. This means that the protected software program can not only verify that it was authorized by the correct publisher, but can also verify that the publisher is certified by the certificate authority.

In a further embodiment, the publisher of the software program can use a toolset to convert the software program into a license-managed product. The certificate Authority certificate is used to issue a publisher certificate to a publisher, and the publisher certificate is used to regulate license terms for using the toolset. The publisher uses the toolset and the publisher certificate to create protected software products and to create product certificates for licensing.

Thus, the present invention provides a chain of certificates to authorize use of a software program through a license. To run, a software product has to verify the product certificates. Verification means verifying the certificate chain, meaning that the product certificate is cryptographically tied to the proper publisher certificate, which in turn, must be cryptographically tied to the certificate authority certificate. The elegance

of the solution is that it allows the certificate authority to control how publishers use the toolset, allows publishers to control how their end-users use their protected software products, and prevents one publisher from authorizing a product from another publisher.

Venkatesan fails to teach or suggest such a software licensing mechanism, and instead teaches a content protection scheme that protects non-executable data. In essence, a relatively large number, n, of identical watermarks is embedded throughout a single software object, through use of n different secret watermark keys. Each of these watermark keys defines a starting location (e.g., in time, space or frequency) in a protected object (or, in a general sense, a pointer to a location in that object) at which a corresponding watermark appears. Once a user has downloaded the protected object through a client computer, the user then transacts with publisher's web server to obtain an electronic license, cryptographically signed by the publisher to an "enforcer" located in that computer, which specifies access rights, which the publisher accords to this client computer, and the watermark value. The client computer contains an enforcer equipped with only one of the n watermark keys. Whenever the client computer attempts to access a file containing a protected object, the enforcer examines the object using its secret watermark key. If the object contains a watermark appearing at a location specified by the enforcer's watermark key, a client operating system accesses a license database to determine whether a signed license made to the enforcer and linked, via the publisher's cryptographic signature, to this protected object resides in that database. A value of a parameter in the license must match a value of the same parameter contained in a watermark detected in the object. In that regard, the license must be signed by the publisher specified in the watermark and made to a product

identification (PID) value that appears in the watermark. Thus, the watermark effectively becomes "glue" between the protected object and its license. If no such license exists, the enforcer inhibits any further access to the object. Otherwise, the enforcer determines whether the watermark value contained in the license matches that detected in the object, and, if so, permits access to the object in accordance with the rights specified in the license. The object can be either an active (executable) or a passive (content) software object. (col. 5, lines 21-57).

Unlike the present invention, in Venkatesan, there is no chaining of certificates. Thus, Venkatesan fails to teach or suggest independent claims 1, 12, 16 and 24. For example, Venkatesan fails to teach or suggest "associating with a software publisher a public and private key pair", as recited in claim 1. Venkatesan teaches that a publisher sets the value of a watermark, and that a watermarking authority (WA) embeds the watermark n times into the object in a starting location determined by corresponding different one of secret keys in order to yield the watermark object. In the present invention, watermarks are not embedded into the protected software product. And in Venkatesan, no public and private key is associated with the publisher.

Similarly, Venkatesan fails to teach or suggest "associating a product public key and private key with a software product", as recited in claim 1. Instead, Venkatesan teaches the use of secret watermark keys that define a starting location in a protected object at which a corresponding watermark appears. Although the publisher may have private/public keys, it is believed that the secret watermark keys are not analogous to the product public and private key pair because the secret watermark keys are not "paired" and different pairs are not associated with different objects. Instead, Venkatesan teaches that "all n watermark keys ... generated by the WA and are

identical across all objects that are to be protected, regardless of their corresponding publishers. These keys are generated once and will be universally used for a relatively long, but finite period, for all objects, from whatever publisher or source, that are to be protected." (Col. 6, lines 2-7).

Because Venkatesan fails to teach associating a product public key and private key with a software product, Venkatesan cannot teach "digitally signing" "at least one of the product private and public keys with the publisher private key", as recited in claim 1.

With respect to generating a license request, Venkatesan may teach that after a user has downloaded a watermarked object, the user, through his(her) client PC, electronically downloads from the publisher's web server an electronic license cryptographically signed by the publisher. However, it is not believed that Venkatesan teaches that "the authorization program" generates the "license request" or that the authorization program "digitally sign[es] the **license request** with the product private key."

Moreover, Venkatesan fails to teach or suggest "generating a license using data **extracted** from the license request and license terms, as recited in claim 1. Venkatesan simply fails to teach that the publisher extracts any data from any license request when returning the electronic license to the user's PC.

In sum, it respectfully submitted that Venkatesan fails to teach or suggest the following combination of elements recited in claim 1:

"associating with a software publisher a public and private key pair" (the publisher certificate);

"associating a product public key and private key with a software product" (the product certificate);

digitally signing "at least one of the product private and public keys with the

publisher private key";

"generating a license request" and "digitally signing the license request with the product private key";

"generating a license" and "signing the license with the publisher private key"; and

"validating the signed license using the publisher public key".

Therefore, it is respectfully submitted that independent claim 1 is allowable over Venkatesan for at least these reasons. Independent claims 12, 16, and 24, include similar recitations and are allowable for lease the same reasons as claim 1.

Turning now to dependent claims 11, 15, and 23, Venkatesan also fails to teach or suggest "preventing use of the software product on a different computer than that used to generate the license request by using a machine fingerprint embedded in the license," In the present invention, all certificates downstream of the certificate Authority certificate are created by a certified (cryptographically signed) license request that originates on the publisher computer in the case of dated the toolset, or on an end-user's user's computer in the case of a protected software program. The license request is generated by the toolset in the case of the publisher and by the protected software program in the case of the end-user's computer. In a further embodiment, because the license request originates on the computer where the toolset/protected software program will be used, the license request can include machine specific information gathered from the computer. The copy protection scheme is the fact that the machine specific information is different for each computer, and can included in the publisher/product certificates in the license request, as recited in claims 11, 15, and 23.

In contrast, Venkatesan asked to use an enforcer to tie a watermark key to a product license. Venkatesan thus has a different copy protection mechanism than the claimed

invention and therefore fails to teach or suggest "embedding a machine fingerprint in the license request."

The arguments above apply with full force and effect to the remaining dependent claims because they are based on allowable independent claims. Therefore, the dependent claims are allowable for at least the same reasons as the independent claims.

In view of the foregoing, it is submitted that claims 1-26 are allowable over the cited references. Because the secondary references stand or fall with the primary references, claims are allowable because they are dependent upon the allowable independent claims. Accordingly, Applicant respectfully requests reconsideration and passage to issue of claims 1-26 as now presented.

Applicants' attorney believes this application in condition for allowance. Should any unresolved issues remain, Examiner is invited to call Applicants' attorney at the telephone number indicated below.

Respectfully submitted,
SAWYER LAW GROUP LLP

October 31, 2005
Date

Stephen G. Sullivan
Attorney for Applicant(s)
Reg. No. 38,329
(650) 493-4540